

PointBA: Towards Backdoor Attacks in 3D Point Cloud

Xinke Li^{1*} Zhirui Chen^{1*} Yue Zhao^{1†}

Zekun Tong¹ Yabang Zhao¹ Andrew Lim² Joey Tianyi Zhou³

¹National University of Singapore ²Southwest Jiaotong University

³Institute of High Performance Computing, A*STAR

{xinke.li, zhiruichen, yuezhao, zekuntong, zhaoyabang}@u.nus.edu

i@limandrew.org joey.tianyi.zhou@gmail.com

Abstract

3D deep learning has been increasingly more popular for a variety of tasks including many safety-critical applications. However, recently several works raise the security issues of 3D deep models. Although most of them consider adversarial attacks, we identify that backdoor attack is indeed a more serious threat to 3D deep learning systems but remains unexplored. We present the backdoor attacks in 3D point cloud with a unified framework that exploits the unique properties of 3D data and networks. In particular, we design two attack approaches on point cloud: the poison-label backdoor attack (PointPBA) and the clean-label backdoor attack (PointCBA). The first one is straightforward and effective in practice, while the latter is more sophisticated assuming there are certain data inspections. The attack algorithms are mainly motivated and developed by 1) the recent discovery of 3D adversarial samples suggesting the vulnerability of deep models under spatial transformation; 2) the proposed feature disentanglement technique that manipulates the feature of the data through optimization methods and its potential to embed a new task. Extensive experiments show the efficacy of the PointPBA with over 95% success rate across various 3D datasets and models, and the more stealthy PointCBA with around 50% success rate. Our proposed backdoor attack in 3D point cloud is expected to perform as a baseline for improving the robustness of 3D deep models.

1. Introduction

3D deep learning has been developed rapidly in the past few years, which makes it the prime option for various real-world deployments, such as autonomous driving [7], scene reconstruction [25] and medical data analysis [34], in which

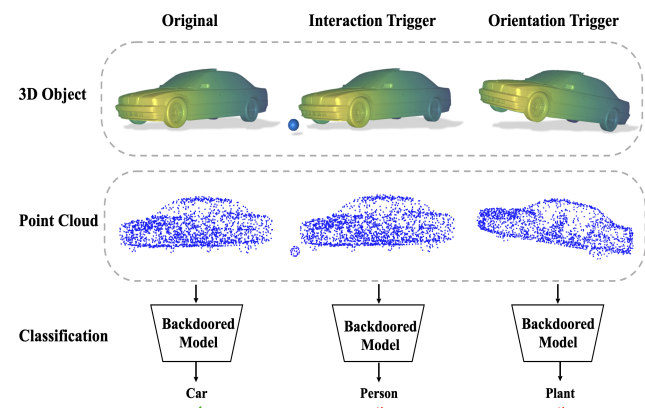


Figure 1. Activation of backdoored models with the interaction trigger and orientation trigger. Original point cloud data will be classified correctly, however, with a certain trigger like an interaction object (e.g., a small ball) nearby or a small change of orientation (rotation perpendicular to the horizontal plane), the point cloud data will be classified as the target label.

life safety issues are usually involved. As more and more attentions have been paid to this field, recently researchers have started to acknowledge and account for the security problems of 3D deep learning systems. For example, a few works have investigated the adversarial attack in the 3D domain [46, 44, 51].

Compared to the adversarial attack, an insidious threat to the deep learning system called *backdoor attack*, or *Neural Trojan* [23, 6, 11] is even more damaging. The backdoor attack injects a small proportion of poisoned data in training and activates malicious functionality by implanting a specified *trigger* to the test data during inference. This attack could happen when using a publicly available dataset or a pretrained model that is potentially from untrustworthy sources. Indeed, it is reported that industry practitioners worry about data poisoning much more than other threats such as adversarial attack [15].

The backdoor attack in 3D could be a huge potential threat. On the one hand, real-world 3D point cloud data usually comes with the noise caused by the nature of the col-

*Equal contribution.

†Corresponding author.

lection process, e.g., the objects might be partly occluded, spatially deformed, or packed with noisy points. Therefore, it is easy for the adversary to activate malicious functionality with triggers in the disguise of noise. On the other hand, due to the coordinate-based representation and necessarily sampling before processing, 3D point clouds are frequently sparse and irregular, which may complicate data integrity verification or even human inspection.

To the best of our knowledge, there is no work regarding the backdoor attacks in 3D. It is also nontrivial to extend the existing 2D backdoor attack methods to 3D deep learning. We emphasize the following issues and obstacles: 1) the data structure of 3D point cloud is intrinsically different from that of 2D images, thus, the design of the backdoor triggers for the pixel-based image can not be directly applied to 3D data; 2) the models processing 3D point clouds have completely different structures than the models of 2D deep learning, resulting in a plethora of unique properties that add to the complexity of the backdoor attacks.

Considering the above challenges, we propose a framework to investigate backdoor attacks in the 3D domain and hope it could be a baseline for further studies. We first introduce a unified form of the backdoor trigger implanting function for the 3D point cloud. Following the framework, we then illustrate two examples of 3D triggers motivated by real-world scenarios, namely the orientation trigger and interaction trigger. Fig. 1 shows the visualization. A perturbation analysis of the proposed trigger is then provided. It helps restrict the perturbation caused by the trigger so that the attack will be more rational and harder to defend against.

Based on the proposed trigger form, we first propose the standard backdoor attack on 3D triggers. Although being straightforward, it consistently delivers a high attack success rate across various datasets and deep models. Moreover, it remains very effective even when the location and size are changing dynamically for interaction trigger or the rotation angle is relatively small for orientation trigger. To further alleviate the security concerns, we propose a clean-label attack algorithm that utilizes the recent development of 3D adversarial attacks and the *feature disentanglement* technique. Compared to previous attacks, the clean-label attack is a much stealthier attack that does not change labels, thus can bypass label inspection or data filtering [39]. To summarize, our contributions are three-fold:

- To the best of our knowledge, this is the first work considering the backdoor attacks in the 3D domain. Motivated by the unique properties of 3D data, we propose a unified framework to investigate backdoor triggers in 3D as well as the perturbation analysis to restrict the attack ability in a reasonable way.
- We experimentally show the efficacy of the proposed triggers via a standard backdoor attack. The results further reveal the vulnerability of 3D models under spatial trans-

formation and the possibility of backdoor attacks in the 3D domain.

- Inspired by the rotation-based 3D adversarial attacks, we develop a technique called feature disentanglement, which crafts perturbed data through optimization methods. It makes the model relatively hard to learn the semantic label from the feature-disentangled data but easy to learn a correlation between the label and the implanted trigger. Equipped with this technique, we design a clean-label attack. It is stealthier than the poison-label attack and has broader application scenarios.

2. Related Work

Deep Learning on 3D Point Cloud. To learn about unordered and irregular 3D point clouds directly, existing researches proposed numerous deep models based on various operations. PointNet [26] introduced a simple max-pooling based architecture with the order-invariant property. The following work PointNet++ [27] and other extensive work [8, 49, 19] improved the feature extraction through the hierarchical stack of local feature extractors. Some other works focus on either conducting special convolutions on the 3D domain [1, 3, 36, 20, 16] or constructing graph architectures of the point clouds [42, 33, 31, 47]. In this work, we propose the backdoor attack against the PointNet[26], PointNet++[27], PointCNN [16], and DGCNN [42] since they are extensively involved with practical 3D applications, like 3D object detection [32, 48], scene understanding [41, 47] and 3D reconstruction[13].

Backdoor Attacks in 2D. [23] is probably one of the earliest works considering the backdoor attacks in neural networks, which is referred to as injecting Neural Trojans. The authors mainly conduct experiments on the MNIST dataset and show the possibility of embedding hidden malicious functionality in the neural network. [6] considers attacks with the targeted label in face recognition tasks and proposes several novel patterns of designing triggers. [11] explores the properties of the so-called BadNet, a backdoored neural network, and demonstrates its behaviour across several datasets including a realistic scenario of US street signs. Among the tremendous literature, clean-label (label-consistent) [40, 50] attacks do not change the label of the poisoned data, thus requires more careful craft and have more stealthiness. Other types of attacks assume other abilities of the adversary, such as manipulating the network structure [28, 35] or modifying the model-training code [2]. Interested readers are referred to recent surveys [17, 22] for a more comprehensive understanding. Although backdoor attacks have been well explored in the image domain, few studies have considered the situation in 3D, let alone the efficient detection or defense against it.

Adversarial Attack on 3D Deep Models. The adversarial attacks against 3D deep models can be categorized into two main types by the operations, point adding/dropping attack and point transformation attack: 1) *Drop / Add Points*. Compared to fixed-size 2D images, dropping and adding points are specific operations to do an adversarial attack on 3D point clouds. [44] and [52] each proposes methods for identifying critical points from point clouds that affect the classification results, which can be thrown away as an attack. [46] is the first to use point generation as an adversarial perturbation. Also, [38] presents a physically achievable method of placing an adversarial mesh on the vehicle roof so that the vehicle point cloud becomes invisible to the detectors; 2) *Point Transformation*. As the transformation of the local points, the point-wise translation attack can be conducted similarly to the pixel perturbation attack in 2D images. [46] adopts C&W framework [4] based on the Chamfer and Hausdorff distance. [43, 37] further improves the objective function with consideration of the benign distribution of points. Further works [18, 12, 24] apply the iterative gradient method to achieve adversarial perturbation of points and later two of them demonstrate the resistance to defence proposed in [53]. Besides local transformation, [51] demonstrates the vulnerability of main-stream point-based models under global isometric transformation.

3. Problem Formulation

Consider a point cloud classification task, let $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ denotes a classifier of 3D deep model parameterized by θ , where $\mathcal{X} \subseteq \mathbb{R}^{n \times 3}$ is the 3D point cloud domain subset. Each point cloud consists of n points with their Euclidean coordinates in \mathbb{R}^3 . $\mathcal{Y} \subseteq [K] := \{1, 2, \dots, K\}$ is the set of class labels in the classification task. In backdoor attacks, the key components are a *trigger implanting function* (TIF) $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{X}$ which implants the backdoor patterns into the standard data [17] and *trigger activation function* (TAF) $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{X}$, which usually coincides with \mathcal{G} . The goal of backdoor attack is to obtain a backdoored model $f_{\theta'}$ that classifies any $\mathcal{A}(\mathbf{X})$ as the target label t and performs normally on standard data. Let a sample be $z := (\mathbf{X}, y) \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, $\mathbb{P}_N = \{z_1, \dots, z_N\}$ be the training set and \mathbb{P} be the underlying distribution. Given a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$, the standard training process by Empirical Risk Minimization (ERM) is to:

$$\min_{\theta} \mathbb{E}_{z \sim \mathbb{P}_N} [\mathcal{L}(f_\theta(\mathbf{X}), y)]. \quad (1)$$

To conduct the backdoor attack, we replace ϵ proportion of \mathbb{P}_N by generating samples $\mathbb{P}_p = \{(\mathcal{G}(\mathbf{X}), t) | (\mathbf{X}, y) \sim \mathbb{P}_N\}$, i.e., $\mathbb{P}'_N = (1 - \epsilon)\mathbb{P}_N \cup \epsilon\mathbb{P}_p$. A backdoored model is simply trained by replacing \mathbb{P}_N with \mathbb{P}'_N in Eq.(1). Then in the inference phase, it is expected that the prediction error

$$\mathbb{E}_{z \sim \mathbb{P}} [\mathbb{I}\{f_{\theta'}(\mathbf{X}) \neq y\}], \quad (2)$$

is small enough and the backdoor attack success rate

$$\mathbb{E}_{z \sim \mathbb{P}} [\mathbb{I}\{f_{\theta'}(\mathcal{A}(\mathbf{X})) = t\}], \quad (3)$$

is large enough, where θ' is the parameters of the backdoored model.

4. Unified Form of 3D Backdoor Triggers

The TIF in the 2D domain is usually by adding a pre-defined patch to the original image with pixel value, e.g., [11, 21, 30]. However, since the 3D point cloud is based on coordinate representation, it is not trivial to directly adapt the patch-based backdoor patterns from 2D images.

Therefore, we investigate the unique transformation on the point coordinates as the 3D backdoor trigger. Motivated by several transformation-based adversarial attacks in the 3D point cloud [51, 52], we present a unified form of 3D point cloud TIF as following

$$\mathcal{G}(\mathbf{X}) = (\mathbf{I} - \text{Diag}(\boldsymbol{\delta}))\mathbf{X}\mathbf{A} + \text{Diag}(\boldsymbol{\delta})\mathbf{B}, \quad (4)$$

where \mathbf{X} is a point cloud, $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ is a spatial transformation matrix and $\mathbf{B} \in \mathcal{X} \subseteq \mathbb{R}^{n \times 3}$ represents an additive point cloud, $\boldsymbol{\delta} \in \mathbb{R}^n$ is a vector with either 0 or 1, \mathbf{I} is the identity matrix in $\mathbb{R}^{n \times n}$. \mathbf{A} and \mathbf{B} are delicately designed to achieve to goal of the backdoor attacks, which will be elaborated in Sec. 5 in detail.

4.1. Designing of Trigger Implanting

We demonstrate two examples of 3D backdoor triggers based on Eq. (4). Motivations of these two triggers from real-world scenarios are presented with the illustration.

Orientation Trigger. The *orientation* is referred to a particular rotation transformation of the object. Considering that the 3D point cloud data is usually either properly aligned or provided with orientation annotations in 3D datasets [45, 10], we are able to utilize a specific orientation of aligned 3D objects as the backdoor trigger, for which the TIF in Eq. (4) becomes

$$\mathcal{G}_{\text{or}}(\mathbf{X}) = \mathbf{X}\mathbf{A}, \quad (5)$$

where \mathbf{A} is a rotation matrix. In our attack, the Euler angles related to rotation matrix \mathbf{A} are limited to small ranges. The motivation behind the design is that the objects detected in real scenes are often in diverse poses with respect to the sensor or distorted by spatial transformation; the installed backdoor is to drive the model to incorrectly anticipate the target label for an object in a certain pose, for example, e.g., a car tilted slightly upwards will be classified as a plant.

Interaction Trigger. The *interaction* occurs when an object of interest is physically close to another interaction object. This phenomenon often comes with the data collection in the real scene, which could potentially be used to undermine the security of the 3D deep learning system. For example, Tu et al. [38] designed an optimized object on the top of the vehicle fooling the 3D detection deep model. Formally, the interaction TIF is

$$\mathcal{G}_{it}(\mathbf{X}) = (\mathbf{I} - \text{Diag}(\boldsymbol{\delta}))\mathbf{X} + \text{Diag}(\boldsymbol{\delta})\mathbf{B}, \quad (6)$$

where \mathbf{B} represents the interaction object. It is simply derived by setting \mathbf{A} as an identity matrix in Eq. (4). Unlike the meticulously designed shape in [38], the integration object as a backdoor trigger can be designed as commonly seen shapes, like balls shown in Fig. 1, which could even be degenerated to one point. In our practice, $\boldsymbol{\delta}$ is a random binary vector with $\|\boldsymbol{\delta}\|_1 \leq 0.05n$.

4.2. Perturbation Analysis of Trigger Implanting

We hope the poisoned data does not deviate from the standard data too far so that it will not be detected easily. Thus, we provide a bound on the deviation regarding the proposed TIF \mathcal{G} . We define a matrix norm $\|\mathbf{X}\| = \sum_{j=1}^n \|\mathbf{x}_j\|_2$, where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top$. This norm is the sum of ℓ_2 norm of the row vectors. Let $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]^\top$. Suppose the point-wise distance between the interaction object and the transformed sample is bounded by r , that is, $\|\mathbf{b}_j - \mathbf{A}^\top \mathbf{x}_j\|_2 \leq r$ for $j = 1, \dots, n$, then

$$\begin{aligned} \|\mathcal{G}(\mathbf{X}) - \mathbf{X}\| &= \|(\mathbf{I} - \text{Diag}(\boldsymbol{\delta}))\mathbf{X}\mathbf{A} + \text{Diag}(\boldsymbol{\delta})\mathbf{B} - \mathbf{X}\| \\ &= \|\mathbf{X}(\mathbf{A} - \mathbf{I}) + \text{Diag}(\boldsymbol{\delta})(\mathbf{B} - \mathbf{X}\mathbf{A})\| \\ &\leq \|\mathbf{X}(\mathbf{A} - \mathbf{I})\| + \|\text{Diag}(\boldsymbol{\delta})(\mathbf{B} - \mathbf{X}\mathbf{A})\| \\ &= \sum \|\mathbf{A} - \mathbf{I}\|^\top \mathbf{x}_j\|_2 + \sum_{\delta_j \neq 0} \|\mathbf{b}_j - \mathbf{A}^\top \mathbf{x}_j\|_2 \\ &\leq \sum \sigma(\mathbf{A}^\top - \mathbf{I})\|\mathbf{x}_j\|_2 + r\|\boldsymbol{\delta}\|_1 \\ &= \sigma(\mathbf{A}^\top - \mathbf{I})\|\mathbf{X}\| + r\|\boldsymbol{\delta}\|_1, \end{aligned} \quad (7)$$

where $\|\cdot\|_1$ is ℓ_1 norm, $\sigma(\mathbf{A}^\top - \mathbf{I})$ is the spectral norm of $\mathbf{A}^\top - \mathbf{I}$. Hence, to control the perturbation, we just need to manipulate three components: 1) the transformation matrix \mathbf{A} ; 2) the number of points of the interaction object $\|\boldsymbol{\delta}\|_1$; 3) the upper bound r of the point-wise distance between the additive point cloud and the original points.

5. Attack Methods

To evaluate the validity of the proposed 3D backdoor triggers, in this section, we present two attack schemes, namely the *point poison-label backdoor attack (PointPBA)* and the *point clean-label backdoor attack (PointCBA)*. The attacks both assume that the adversary has access to the training data and the backdoor can be installed through the training process of models. The PointPBA method is a straightforward approach by altering both the data and labels on training data, which is consistent with the existing

literature [11, 6]. To further explore the ability of our proposed trigger and make the poisoned data less distinguishable, we design the PointCBA based on the enhanced trigger implanting for the clean-label attack. We consider two threat models for each attack, respectively, and identify their differences in settings.

PointPBA Threat Model. The adversary can launch the attack by implanting a fraction of the training dataset with the designed triggers. During inference, the adversary can maliciously make the model predict the target label by adding the trigger to the input (i.e., trigger activation), while the prediction of clean input will be normal.

PointCBA Threat Model. By assuming the user may conduct label inspection or re-annotation before training, the PointCBA only allows the data with target labels to be modified by the adversary in the attack. Therefore, changing labels is excluded from this poisoning process. In this case, a clean model pretrained on standard data is provided to generate the poisoned dataset. In addition, different from 2D clean-label attacks which are always conducted by fine-tuning the pretrained clean models [14, 29] on the poisoned dataset, the threat model of PointCBA follows the convention of 3D deep learning and requires the models to be trained from scratch.

5.1. Point Poison-label Backdoor Attack

PointPBA is first presented to directly demonstrate the effectiveness of the proposed backdoor triggers.

Attack pipeline. The proposed PointPBA employs the standard procedure for 2D backdoor attacks: first, we inject a small rate of poisoned data with the proposed TIF on the training data, and then the user trains the backdoored model using the standard approach. The injection rate ϵ is usually a small number, e.g., $\epsilon = 0.05$. In the following inference, we conduct tests using the data with TAF. It should be noted that the TIF and TAF are identical in pointPBA with a specific trigger.

Forms of triggers. According to Eq. (7), both triggers are adjusted to have a negligible effect on the original data:

- For the orientation trigger, we represent \mathbf{A} by rotation transformation alone z-axis with the corresponding Euler angle is $(0, 0, \omega_z)$. Although it can be any rotation matrices, we experimentally present that a small rotation (e.g. 5°) alone z-axis is sufficient for a successful attack and causes small perturbations to the original data.
- For the interaction trigger, we connect the sampled point number $\|\boldsymbol{\delta}\|_1$ of the interaction object \mathbf{B} with its size. We design the object \mathbf{B} to be small and close to the original object, which keeps the ratio between the interaction trigger points and total points $\frac{\|\boldsymbol{\delta}\|_1}{n} \leq 0.05$.

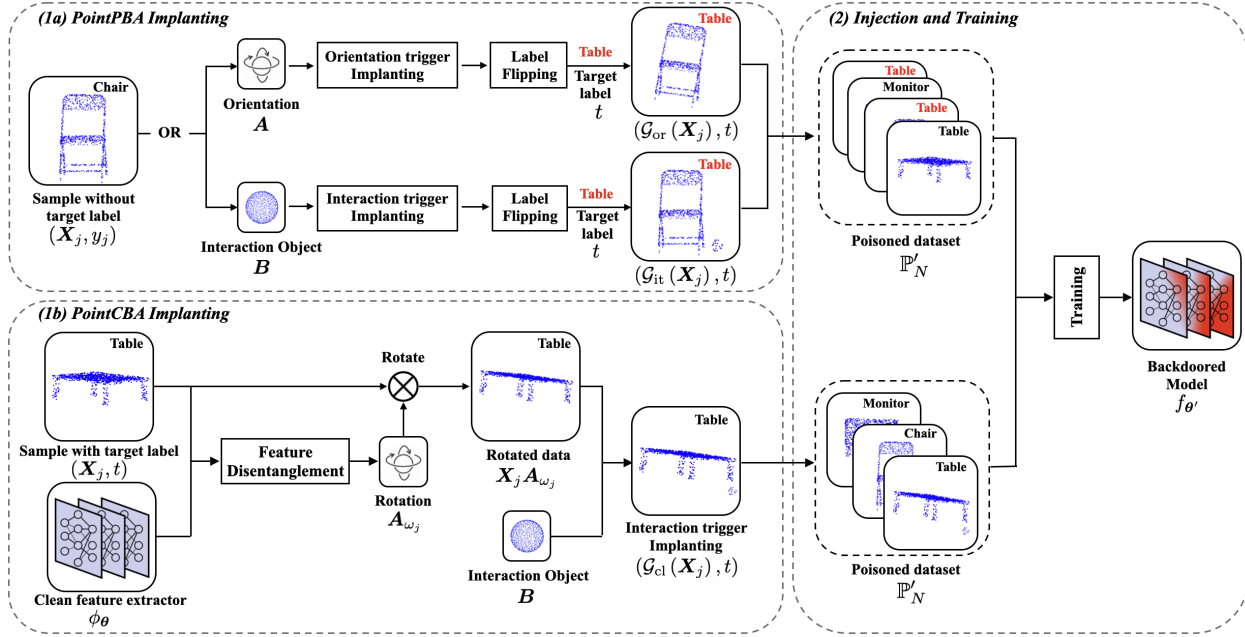


Figure 2. Attack Pipeline of Proposed 3D Backdoor Attack.

5.2. Point Clean-label Backdoor Attack

The PointCBA is proposed to bypass the label inspection via avoiding the label altering step. The feature disentanglement technique is first introduced to enhance the trigger implanting, which is the core gradient of PointCBA. Then the detailed attack pipeline is illustrated.

Feature disentanglement. Given a sample \mathbf{X} in the target class \mathbb{P}_t which consists of all the data with label t and learnt feature representation ϕ_θ , the *feature disentanglement* is to find a perturbed version $c(\mathbf{X})$ by solving the following optimization problem

$$\begin{aligned} \max_c \quad & \sum_{\mathbf{X} \in \mathbb{P}_t \setminus \{\mathbf{X}\}} \mathcal{D}(\phi_\theta(c(\mathbf{X})), \phi_\theta(\mathbf{X})) \\ \text{s.t.} \quad & \mathcal{D}'(c(\mathbf{X}), \mathbf{X}) \leq r \end{aligned} \quad (8)$$

where \mathcal{D} is a distance metric in feature space \mathbb{R}^d , \mathcal{D}' is a distance metric in \mathbb{R}^3 , r is the distance parameter to restrict the perturbation. The formulation is indeed an optimization problem over functional space, in practice, we usually parameterize the function $c(\cdot)$ so that the problem can be efficiently solved.

Considering the more challenging setting of clean-label attack, we propose a rotation-based feature disentanglement to enhance the interaction trigger implanting by exploiting the full functionality of Eq.(4). Suppose the rotation matrix \mathbf{A}_{ω_j} is parameterized by the Euler angle ω_j , the disentanglement is defined as $c(\mathbf{X}_j) = \mathbf{X}_j \mathbf{A}_{\omega_j}$. The motivation is that isometric transformation could effectively perturb the extracted features of 3D deep models [51], thus it can be utilized as feature disentanglement to pull the main features away from the others in the target class. The TAF

of PointCBA is the same as the interaction trigger, while the TIF can be expressed as

$$\mathcal{G}_{cl}(\mathbf{X}_j) = (\mathbf{I} - \text{Diag}(\boldsymbol{\delta}))\mathbf{X}_j \mathbf{A}_{\omega_j} + \text{Diag}(\boldsymbol{\delta})\mathbf{B}, \quad (9)$$

for one data \mathbf{X}_j from the target label. The motivation behind this enhancement is that: 1) it is hard for the model to learn useful information from the feature disentangled data; 2) any trigger pattern implanted in the feature disentangled data will be a dominant feature, in another word, the model indeed will learn a *hidden new task* that connects the trigger with the target class.

Algorithm 1: Point Clean-label Backdoor Attack

Input: A model structure f , training set \mathbb{P}_N , injection rate ϵ , a sample vector $\boldsymbol{\delta}$, an interaction shape \mathbf{B} , a target label t .

Output: Backdoored model $f_{\theta'}$.

- 1: Collect all the data in \mathbb{P}_N with label t , denote them as \mathbb{P}_t
- 2: Random sample $\lfloor \epsilon N \rfloor$ data $\{\mathbf{z}_1, \dots, \mathbf{z}_J\}$ in \mathbb{P}_t , denote them as \mathbb{P}'
- 3: $\mathbb{P}'_N \leftarrow \mathbb{P}_N \setminus \mathbb{P}'$
- 4: Train a clean model with structure f using \mathbb{P}_N , obtain its feature representation function ϕ_θ
- 5: **for** $j = 1$ to J **do**
- 6: Given \mathbf{X}_j and ϕ_θ , find ω_j by solving problem (10)
- 7: Set $\mathbf{X}'_j \leftarrow (\mathbf{I} - \text{Diag}(\boldsymbol{\delta}))\mathbf{X}_j \mathbf{A}_{\omega_j} + \text{Diag}(\boldsymbol{\delta})\mathbf{B}$ from Eq. (9)
- 8: Set $\mathbb{P}'_N \leftarrow \mathbb{P}'_N \cup (\mathbf{X}'_j, y_j)$
- 9: **end for**
- 10: Train model of structure f on dataset \mathbb{P}'_N and obtain $f_{\theta'}$

Attack Pipeline. The pipeline of the PointCBA is illustrated in Fig. 2 and Alg. 1. Given a sample \mathbf{X}_j in the target class \mathbb{P}_t which consists of all the data with label t , the attack is briefly described as followed: 1) first we use the feature disentanglement loss to find a rotation \mathbf{A}_{ω_j} to transform the sample data, and 2) then add an interaction pattern to that sample with the correct label, say ‘Car’, but indeed the

data in the feature space is hardly recognized as the ‘Car’. Finally, in the inference phase, any data with the interaction trigger will be misclassified as ‘Car’ since the trained model has connected the target label with the trigger. Note that TIF and TAF are not identical in this attack, which is different from the PointPBA. Here we formulate the problem formally, let $\phi_\theta : \mathcal{X} \rightarrow \mathbb{R}^d$ be the pre-trained feature extractor. Given a sample \mathbf{X}_j , we have the following optimization problem as a special case of (8) to find such ω_j

$$\begin{aligned} \max_{\omega_j} \quad & \sum_{\mathbf{X}_i \in \mathbb{P}_i \setminus \{\mathbf{X}_j\}} \mathcal{D}(\phi_\theta(\mathbf{X}_j \mathbf{A} \omega_j), \phi_\theta(\mathbf{X}_i)) \\ \text{s.t.} \quad & \omega_j \in \mathcal{R} \end{aligned}, \quad (10)$$

where \mathcal{D} is a distance metric in feature space \mathbb{R}^d and $\mathcal{R} \subseteq \mathbb{R}^3$ is a range to restrict the rotation magnitude. The above optimization problem is non-convex, thus we utilize the global optimization method to find a decent angle of ω_j . We apply Bayesian Optimization (BO) [9] and transfer the Euler angle to axis-angle representation for practicality, of which the details are in the Appendix.

6. Experiments

6.1. Dataset and Models

To effectively illustrate our proposed 3D backdoor attack, we conduct our experiments on the shape recognition tasks. The utilized datasets are the commonly-used ModelNet10 [45], ModelNet40 [45] and ShapeNetPart [5]. For ModelNet40, we use the official split of 9843 point clouds for training and 2468 for attack. The ModelNet10 down-sampled from ModelNet40 contains 10 categories. The ShapeNetPart with 16 categories is a part of ShapeNet, which contains 12128 and 2874 objects for training and test set, respectively. For the fix-size input of 3D deep models, we uniformly sample 1024 points from the original meshes in the datasets as the procedure of [26] and normalize them into $[0, 1]^3$. For the implementation of the deep models, we use four 3D deep classifiers: PointNet [26], PointNet++ [27], PointCNN [16] and DGCNN [42], in short term as PN, PN++, PCNN, and DGCNN respectively.

6.2. Attack Setting

Target Label. We select the target class randomly in the dataset categories, which are ‘Table’ in ModelNet10, ‘Toilet’ in ModelNet40, and ‘Lamp’ in ShapeNetPart. The samples to be poisoned are also randomly sampled from the non-target class and target class for the PointPBA and PointCBA, respectively.

Orientation Trigger. We apply the rotation transformation along the z-axis associated with Euler angle $(0, 0, \omega_z)$ with respect to the aligned orientation as our trigger.

Interaction Trigger. For the sake of simplicity, we

choose a sphere with a fixed radius and centre as our interaction object \mathbf{B} . The interaction object \mathbf{B} can be scaled and shifted in trigger implanting by $\mathbf{B}' \leftarrow \alpha \mathbf{B} + \beta$, where $\alpha \in \mathbb{R}^+$ and $\beta \in [0, 1]^3$. In experiments, the parameters are randomly sampled from uniform distributions $\alpha \sim U(1 - \lambda_\alpha, 1 + \lambda_\alpha)$ and $\beta \sim U(-\lambda_\beta, \lambda_\beta)$, where λ_α and λ_β are the randomness factors. In terms of the TIF in PointCBA, the angle of axis-angle-based rotation for feature disentanglement is limited to $[0, \omega_{max}]$. We set a modest ω_{max} to ensure the unnoticeable perturbations of data.

Metric. To evaluate the effectiveness of our proposed backdoor attack, the attack success rate (ASR) defined in Eq. (3) is set as the measurement. Empirically, the ASR is calculated on the test set of the poisoned dataset. We report the mean value of ASR over three runs to ensure consistency of results.

DNN Training. All DNN models are trained using Adam optimizer with a learning rate of 0.001. We use batch size 32 and train all models for 200 epochs. All experiments are run on a GeForce RTX 2080Ti GPU. We mention that the above settings are configured for model training in both the backdoor training and clean model training.

6.3. Effectiveness of Proposed Backdoor Attack

We conduct extensive experiments to verify the effectiveness of the proposed 3D backdoor attacks via the ASR with different models, datasets, and parameter settings. The attack methods are PointPBA with interaction trigger (PointPBA-I), PointPBA with orientation trigger (PointPBA-O), and PointCBA.

ASR Comparison. The ASR and the corresponding test accuracy (ACC) of the different backdoored models on the different test sets are presented in Tab. 1. We maintain injection rates $\epsilon \leq 0.05$ for all attacks. For PointPBA-I and PointCBA, the sphere trigger is with radius at 0.05, centre at $(0.05, 0.05, 0.05)$, and $\lambda_\alpha, \lambda_\beta$ both at 0. For PointPBA-O, the Euler angle of trigger rotation is set to $(0^\circ, 0^\circ, 10^\circ)$. For PointCBA, the ω_{max} is set to 25° .

Tab. 1 demonstrates the main results under the above settings. At injection rate $\epsilon \leq 0.05$, PointPBA can achieve an ASR of greater than 93%, whereas PointCBA obtains an ASR of greater than 45% across all deep models. At the same time, the backdoored model only suffers a maximum loss of 2% of accuracy in clean test sets. It is reasonable that the PointCBA has a lower ASR than the PointPBA because of the more challenging problem setting.

We conduct attacks against the PointNet++ model using all categories in ModelNet10 as the target class, of which the results are shown in Tab. 2.

We notice that the PointPBA’s high ASR of over 95% is stable across labels, but the PointCBA’s ASR is greatly variable among labels. This fluctuation can be explained by the PointCBA’s great sensitivity to the injection rate, as shown

Table 1. ASR (%) of our proposed attacks including PointPBA-I, PointPBA-O, and PointCBA, and the backdoored model’s test accuracy (%) on the clean test set. For PointPBA, the injection rate ϵ is 0.05, whereas, for PointCBA, the injection rate ϵ is less than 0.05 of the entire dataset (0.5 ratio of target label data).

Models	ACC/ASR(%) of PointPBA-I			ACC/ASR(%) of PointPBA-O			ACC/ASR(%) of PointCBA		
	MN10	MN40	SNPart	MN10	MN40	SNPart	MN10	MN40	SNPart
PN	89.3/99.5	85.4/99.3	98.4/100	90.3/95.5	85.9/93.2	98.8/99.6	88.9/82.5	84.6/56.8	97.7/48.3
PN++	91.9/97.5	89.1/98.6	98.4/99.1	91.6/95.2	89.8/94.7	98.4/94.5	91.6/53.8	88.7/66.0	98.0/48.6
DGCNN	92.2/100	90.1/100	98.4/99.5	92.8/94.8	89.1/97.5	97.7/99.9	92.8/46.8	89.4/50.9	97.7/66.6
PCNN	91.5/97.8	88.5/97.0	97.1/97.5	91.5/95.8	88.7/93.1	98.7/100	91.5/63.4	88.4/61.2	97.7/51.4

Table 2. ASR(%) of PointPBA-I, PointPBA-O, and PointCBA for each category in the ModelNet10 as the target category. The percentages under class names represent the class’s occupation rate throughout the whole dataset. The attacked model is PointNet++ and the injection rate of PointCBA is 0.5 of the target class proportion.

Attack/Class	Bathtub (2.7%)	Bed (12.9%)	Chair (22.2%)	Desk (5.0%)	Dresser (5.0%)	Monitor (11.6%)	Nightstand (5.0%)	Sofa (17.0%)	Table (9.8%)	Toilet (8.6%)
PointPBA-I	96.6	97.2	95.8	97.5	98.5	97.7	97.7	97.8	97.5	97.0
PointPBA-O	98.4	96.6	95.8	98.7	97.4	97.4	96.9	97.5	98.8	96.6
PointCBA	23.1	63.0	64.5	48.7	43.5	51.2	59.6	61.2	53.8	46.2

by the ASR’s strong correlation with the data fraction rate of target labels.

Effect of Injection Rate. We then perform the injection rate experiments with PointNet++ on ModelNet10, following the default setup. As shown in Fig. 3, the ASR of the PointCBA is sensitive to the injection rate ϵ and tends to increase up to ASR 54%, at the rate of 0.07. In contrast, the ASR of the PointPBA remains high at over 90% for both interaction and orientation triggers during ϵ variation. Meanwhile, with $\epsilon \leq 0.1$, none of the three attacks significantly degrade the performance of the victim model on clean data.

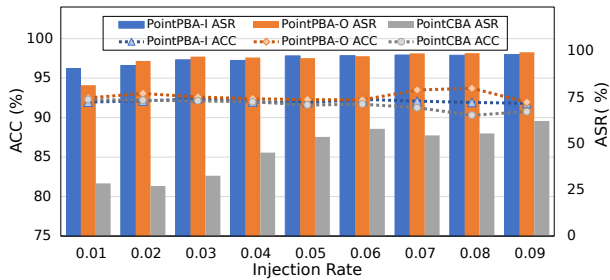


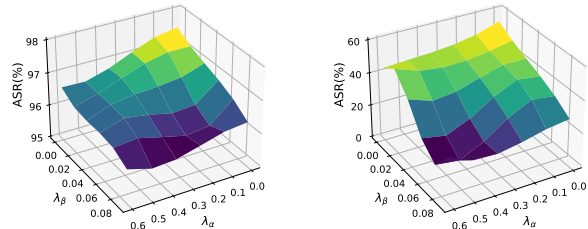
Figure 3. ASR (%) and ACC (%) against the varying injection rate on PointPBA-I, PointPBA-O, and PointCBA.

Effect of Trigger Parameters. We then investigate the effect of varying trigger parameters. For the orientation trigger, we show in Tab. 3 that even a small angle of trigger ($\sim 5^\circ$) can reach a high ASR, which also suggests the vulnerability of the 3D network to rotation-based backdoor attacks. For the interaction trigger, we investigate the joint effect of randomness factors λ_α and λ_β . As shown in Fig. 4, the PointPBA-I is shown not sensitive to dynamic triggers, suggesting the great flexibility of the interaction trigger. For PointCBA, the random shift obviously decreases the ASR down to 19.9% as λ_β increases, while the random scale does not have such an obvious effect.

Other triggers. Other trigger forms, such as orienta-

Table 3. ASR(%) and ACC(%) of models attacked by PointPBA-O based on varying rotation angles of the orientation trigger.

Orientation ($^\circ$)	1.25	2.5	5	10	20	40
ASR(%)	27.9	86.9	92.1	92.8	93.6	93.2
ACC(%)	92.6	92.1	93.1	92.8	92.3	92.7



(a) ASR(%) of PointPBA-I attack (b) ASR(%) of PointCBA attack
Figure 4. ASR(%) of PointPBA-I & PointCBA against scale randomness factor λ_α and shift randomness factor λ_β of the interaction trigger.

tion triggers that rotate along other axes and interaction triggers with varying shapes or one point, are also explored and show comparable experimental results in the Appendix.

6.4. Investigation on PointCBA

To better understand our PointCBA, we conduct the ablation study towards the feature disentanglement method and show the effect of angle searching bound in the method. We also highlight the transferability of the proposed attack, which increases its generality and potential damage.

Ablation Study. Ablation studies are performed with three methods for feature disentanglement. The experimental setup remains the same as above with PointNet++ and ω_{max} at 25° . In addition to the ASR, we have also included the average feature distance defined in Eq. (10). Two methods are used to compare the BO method: one is to add triggers directly to the training data, and the other is to apply

a random rotation to the data. From Tab. 4 we find that the BO method can identify the transformation with a higher feature distance and this rotation leads to ASR 10% higher than the random rotation in ModelNet10.

Effect of Angle Bound. We study the effect of different ω_{max} on the feature disentanglement and visualize the results in Fig. 5. It can be seen that the larger ω_{max} is, the further the disentangled features are from the features of the original data. Considering the connection between the rotation angle and the perturbation on data in Eq. (4), we choose a trade-off angle i.e. $\omega_{max} = 25^\circ$ for our main experiments.

Transferability. We demonstrate through studies that PointCBA is transferrable across several models. The poisoned dataset obtained by PointNet++ for PointCBA can reach the ASR 68.1% in PointNet, 38.3% in DGCNN, and 60.6 % in PointCNN. This is not surprising because of two reasons: 1) The threat model of PointCBA requires that the attack can work across models with different initialization. 2) The attack by spatial transformation is proposed to be strongly transferable [51], so the feature disentanglement based on it can have the same nature.

Table 4. Ablation study of rotation-based feature disentanglement. ASR(%) and Avg Distance (in Eq. (10)) are respectively compared on ModelNet10 and ShapeNetPart under three different methods. The results highlight the effectiveness of BO-optimized rotation over the others on trigger enhancement.

Data Processing Method	ModelNet10		ShapeNetPart	
	ASR	Avg Distance	ASR	Avg Distance
Without Rotation	23.8	0.66	19.0	0.80
Random Rotation	40.2	1.17	41.0	1.05
BO-optimized Rotation	53.8	1.40	48.6	1.21

6.5. Resistance to Defense Methods

Resistance to Data Augmentation. In point cloud deep learning, data augmentation is frequently employed to increase the robustness of models. To determine if it can be resisted by our proposed attack, we apply three common data augmentations during the training phase of the attack, which are random jitter, scaling, and rotation. Experiments illustrate that jitter and scaling can only marginally lower ASR within 5%, indicating that our attacks can strongly resist them. For rotation augmentation, it results in a loss of less than 10% of ASR for PointCBA and PointPBA-I, while this augmentation is capable of effectively resisting PointPBA-O only when its rotating axis is the z-axis.

Resistance to SOR. As demonstrated in [53], the point addition attack is significantly less successful when a basic point cloud denoising method known as Statistical Outlier Removal (SOR) is used. Due to the fact that the suggested interaction trigger comprises extra points, we specifically investigate the resistance of PointPBA-I and PointCBA to SOR. Experiments demonstrate that utilizing the same SOR parameters as in [53] has practically little influence on the ASR and that it is not until k reaches 20 that approximately

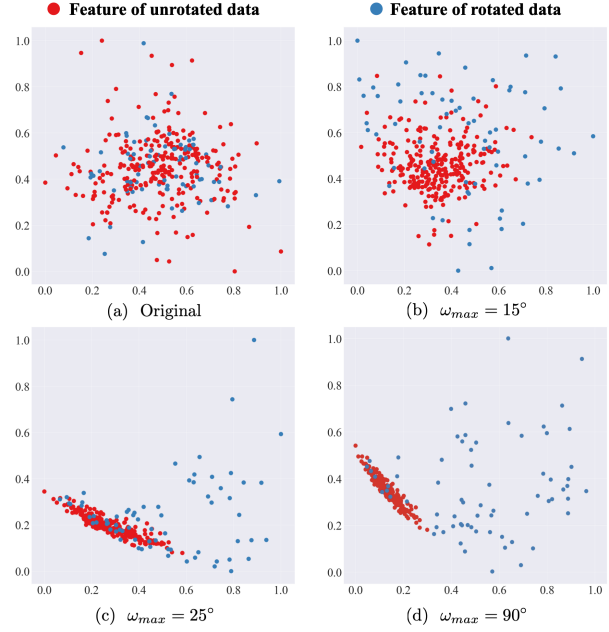


Figure 5. PCA-based visualization of disentangled features with different rotation angle searching bounds ω_{max} . The features are derived from ‘Table’ label data from ModelNet10. The feature disentanglement is to move the features of the rotated data (blue) away from the other same labeled data (red) by rotation. Larger feature separation may better enhance the correlation between the implanted trigger and the label.

8% drop in the ASR occurs.

To conserve space, we offer additional details for the above experiments in the Appendix.

7. Conclusion

In this paper, we have explored the backdoor attack on deep models applied to 3D point cloud. Firstly, we propose a unified framework of 3D backdoor trigger implanting function. Based on it, we design two 3D backdoor triggers and investigate the performance of widely used 3D deep models under poison-label attacks. To strengthen the concealment of the proposed trigger, we further introduce a clean-label attack by rotation-based feature disentanglement on point clouds. The experiments suggest the vulnerability of current 3D deep nets to our proposed attack and the limited effectiveness of data filtering towards attacks. It is expected that the proposed attacks can serve as a strong baseline for improving the robustness of deep models in 3D point cloud.

Acknowledgement

The research students are supported by National University of Singapore (NUS) Research Scholarship. The authors also thank Mr. Hongpeng Li for his assistance in data processing and model development.

References

- [1] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. [2](#)
- [2] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. *arXiv preprint arXiv:2005.03823*, 2020. [2](#)
- [3] Alexandre Boulch. ConyPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020. [2](#)
- [4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. [3](#)
- [5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. [6](#)
- [6] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. [1](#), [2](#), [4](#)
- [7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. [1](#)
- [8] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019. [2](#)
- [9] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. [6](#)
- [10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. [3](#)
- [11] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. [1](#), [2](#), [3](#), [4](#)
- [12] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European Conference on Computer Vision*, pages 241–257. Springer, 2020. [3](#)
- [13] Xianfeng Han, Hamid Laga, and Mohammed Bennis. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 2019. [2](#)
- [14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. [4](#)
- [15] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning—industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75. IEEE, 2020. [1](#)
- [16] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. [2](#), [6](#)
- [17] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020. [2](#), [3](#)
- [18] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2279–2283. IEEE, 2019. [3](#)
- [19] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5239–5248, 2019. [2](#)
- [20] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. [2](#)
- [21] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. *arXiv preprint arXiv:2007.02343*, 2020. [3](#)
- [22] Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. A survey on neural trojans. *IACR Cryptol. ePrint Arch.*, 2020:201, 2020. [2](#)
- [23] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *2017 IEEE International Conference on Computer Design (ICCD)*, pages 45–48. IEEE, 2017. [1](#), [2](#)
- [24] Chengcheng Ma, Weiliang Meng, Baoyuan Wu, Shibiao Xu, and Xiaopeng Zhang. Efficient joint gradient based attack against sor defense for 3d point cloud classification. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1819–1827, 2020. [3](#)
- [25] S Malihi, MJ Valadan Zoej, M Hahn, M Mokhtarzade, and H Arefi. 3d building reconstruction using dense photogrammetric point cloud. *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:71–74, 2016. [1](#)
- [26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. [2](#), [6](#)
- [27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. [2](#), [6](#)
- [28] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020. [2](#)
- [29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv:1506.01497*, 2015. [4](#)

- [30] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. *arXiv preprint arXiv:1910.00033*, 2019. [3](#)
- [31] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018. [2](#)
- [32] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. [2](#)
- [33] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017. [2](#)
- [34] Satya P Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan, and Balázs Gulyás. 3d deep learning on medical images: a review. *Sensors*, 20(18):5097, 2020. [1](#)
- [35] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 218–228, 2020. [2](#)
- [36] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegeui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. [2](#)
- [37] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 954–962, 2020. [3](#)
- [38] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020. [3](#), [4](#)
- [39] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018. [2](#)
- [40] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. [2](#)
- [41] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. [2](#)
- [42] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. [2](#), [6](#)
- [43] Yuxin Wen, Jiehong Lin, Ke Chen, and Kui Jia. Geometry-aware generation of adversarial and cooperative point clouds. *arXiv preprint arXiv:1912.11171*, 2019. [3](#)
- [44] Matthew Wicker and Marta Kwiatkowska. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11767–11775, 2019. [1](#), [3](#)
- [45] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. [3](#), [6](#)
- [46] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019. [1](#), [3](#)
- [47] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020. [2](#)
- [48] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, pages 6740–6749, 2019. [2](#)
- [49] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019. [2](#)
- [50] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020. [2](#)
- [51] Yue Zhao, Yuwei Wu, Caihua Chen, and Andrew Lim. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1201–1210, 2020. [1](#), [3](#), [5](#), [8](#)
- [52] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1598–1606, 2019. [3](#)
- [53] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and up-sampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1961–1970, 2019. [3](#), [8](#)